Automatic aerial target detection and tracking system in airborne FLIR images based on efficient target trajectory filtering

Carlos R. del-Blanco[a], Fernando Jaureguizar[b], Luis Salgado[c] and Narciso García[d]

Grupo de Tratamiento de Imágenes, Universidad Politécnica de Madrid, 28040, Madrid, Spain

ABSTRACT

Common strategies for detection and tracking of aerial moving targets in airborne Forward-Looking Infrared (FLIR) images offer accurate results in images composed by a non-textured sky. However, when cloud and earth regions appear in the image sequence, those strategies result in an over-detection that increases very significantly the false alarm rate. Besides, the airborne camera induces a global motion in the image sequence that complicates even more detection and tracking tasks. In this work, an automatic detection and tracking system with an innovative and efficient target trajectory filtering is presented. It robustly compensates the global motion to accurately detect and track potential aerial targets. Their trajectories are analyzed by a curve fitting technique to reliably validate real targets. This strategy allows to filter false targets with stationary or erratic trajectories. The proposed system makes special emphasis in the use of low complexity video analysis techniques to achieve real-time operation. Experimental results using real FLIR sequences show a dramatic reduction of the false alarm rate, while maintaining the detection rate.

Keywords: Target Detection, Tracking, FLIR images, Trajectory Analysis, Ego-motion

1. INTRODUCTION

An automatic target detection and tracking system (ATDT) focused on aerial targets for FLIR imagery, acquired in a airborne camera, is addressed in this paper. This scenario presents several difficulties such as unpredictable camera motion (called global motion or ego-motion), special characteristics of aerial images, and the reduced size and motion of aerial targets. Camera ego-motion distorts spatiotemporal correlation between consecutive images, increasing the difficulty of tracking and detection tasks based on motion strategies. On the other hand, the background of aerial images is mainly composed by non-textured sky regions, also cloud and earth regions appear, which are sources of static and dynamic clutter that can be detected as targets. Besides, the reduced size and motion of aerial targets, because of the long distance between camera and targets, reduces the detection performance and increases the false alarm rate.

Most of the works address each task, detection and tracking, independently along with a method to compensate or limit the ego-motion effect. In [1]-[2] motion-based segmentation schemes with a previous ego-motion compensation stage are presented, but their approach tends to produce an over-detection. Besides, the high complexity of the ego-motion compensation approaches used in these works limits their applicability in real-time systems. To solve this problem, [3] employs a multi-resolution pyramid to reduce the computational burden. In [4]-[5] similar ego-motion compensation schemes are used when target tracking fails because of large camera displacements, but there is not any feedback to the detection stage. Others works, as in [6], expand the target search area in the tracking process to alleviate the camera ego-motion without using any explicit compensation technique. But, this approach lacks flexibility and its computational burden increases with the size of the search area.

In this paper an ATDT system is presented, which achieves exceptional results by refining the detection process through a novel target trajectory filtering strategy. Moreover, it keeps real time requirement by means of the use of a low complexity motion compensation technique. The system estimates local motion through a fast selective-search block-matching algorithm, which is used to infer global motion by means of a robust

[a]cda@gti.ssr.upm.es, [b]fjn@gti.ssr.upm.es, [c]L.Salgado@gti.ssr.upm.es, [d]narciso@gti.ssr.upm.es; www.gti.ssr.upm.es

Automatic Target Recognition XVII, edited by Firooz A. Sadjadi, Proc. of SPIE Vol. 6566, 656604, (2007) · 0277-786X/07/$18 · doi: 10.1117/12.718582

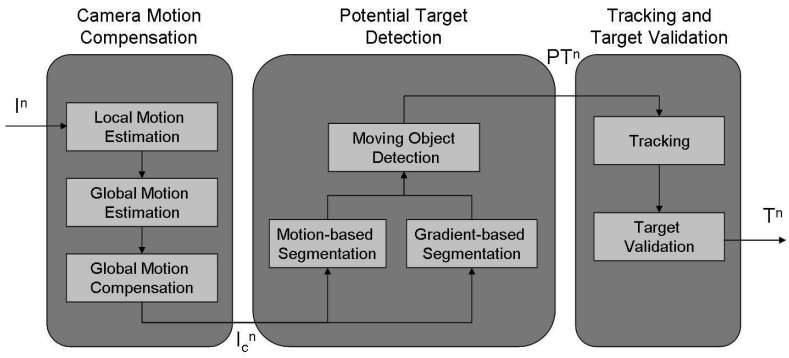


Figure 1. System block diagram.

parameter estimation technique that assumes a restricted-affine motion model [7]. The potential aerial targets are detected through a combination of gradient and motion segmentation techniques, applied on the global motion compensated image sequence. Potential targets are tracked using a template correlation technique and a Kalman filter. Finally, each potential target trajectory is analyzed through a robust curve fitting technique, filtering those that show no motion o erratic trajectories. Experimental results using real FLIR sequences demonstrate the efficiency of this strategy, reaching a very low false alarm rate.

This paper has been organized as follows: in Section 2 the ATDT system is described, and each one of its processing modules is presented independently; experimental results with real FLIR images are presented in Section 3; and finally conclusions are exposed in Section 4.

2. SYSTEM DESCRIPTION

The system is composed by three main processing modules: *Camera Motion Compensation*, *Potential Target Detection* and *Tracking and Target Validation*. In short, the description of the system (Fig. 1) is as follows: FLIR input image sequence is processed by the *Camera Motion Compensation* module, which accurately estimates a non-dense motion vector field between consecutive images by means of a fast selective-search block-matching algorithm. This motion vector field is used to robustly estimate global motion parameters related to the camera, assuming a restricted affine motion model. Global motion is compensated warping the image according to the parameters of the motion model previously computed. *Potential Target Detection* module performs a motion-based segmentation in parallel with an edge-based segmentation, which are combined to obtain a set of potential targets. The *Tracking and Target Validation* module tracks every potential target by means of a template correlation technique. Resulting trajectories are analyzed using a robust curve fitting technique that filters false moving targets with erratic or non-linear trajectories and validates the rest of targets.

2.1. Camera Motion Compensation

This module stabilizes the sequence from the unpredictable camera motion, called ego-motion, that arises from the airborne camera. It is divided into three stages: *Local Motion Estimation*, *Global Motion Estimation* and *Global Motion Compensation*. In the first stage, local motion is estimated from consecutive images, I^{n-1} and I^n, to yield a non-dense motion vector field ($MVF^{n-1,n}$) through a low complexity block matching algorithm.

The *Global Motion Estimation* stage infers global motion, corresponding to the camera, from $MVF^{n-1,n}$ using a restricted-affine motion model along with RANSAC [8], a robust parameter estimator. As a result, a parameter matrix that defines the camera motion is obtained, which is used in the Global Motion Compensation stage to warp I^n, stabilizing it respect to I^{n-1}.

2.1.1. Local Motion Estimation

Local motion is estimated from consecutive images, I^{n-1} and I^n, using a diamond-pattern search block-matching algorithm [9], adapted to compute only blocks with high reliable motion information.

Block-matching algorithm divides the current image, I^n, into a matrix of non-overlapped square blocks which are compared with the corresponding blocks in the previous image, I^{n-1}, and its adjacent neighbors located inside a predefined search area, S_a. The best matching between blocks produces a motion vector that defines the movement of one block in I^n with the corresponding one in I^{n-1}. The set of estimated motion vectors related to all the blocks of I^n constitutes the motion vector field, $MVF^{n-1,n}$, that defines the local motion between I^{n-1} and I^n. Search area S_a is constrained to a rectangle surrounding the corresponding block in I^{n-1}. The best matching is computed minimizing the mean absolute difference cost function (MAD), whose expression is given in (1):

$$MAD(d_x, d_y) = \frac{1}{N^2} \sum_{(x,y)\in B} \left| I^n(x,y) - I^{n-1}(x+d_x, y+d_y) \right| \tag{1}$$

where B is a $N \times N$ block and (d_x, d_y) are the coordinates of each candidate motion vector inside S_a.

To reduce the number of MAD computations, a diamond-pattern search method (DPS) [9] has been used, instead of an exhaustive search one, allowing to reduce significantly the computational burden. The DPS method starts evaluating nine locations in S_a which follows a diamond-shape pattern (9DP), see Fig. 2(a), where the central location is in the centre of S_a. If the location of minimum MAD is not the centrum of the pattern, then the 9DP is displaced to the location of the minimum MAD, and a new minimum is searched. Since some locations have just been evaluated before, only is necessary to compute 3 o 5 new locations depending on the position of the minimum. This process is repeated until the minimum MAD is associated to the central location of the 9DP. Once this is accomplished, a five location diamond-shape pattern (5DP) is used, see Fig. 2(b). Similarly to the 9DP, the process is repeated until the minimum MAD corresponds to the central location, taking into account not to compute previous locations. When the minimum MAD is the central location, the search finishes, and its coordinates are used to generate the corresponding motion vector. In Fig. 2(c) the entire search method is outlined.

Although in a standard block-matching algorithm each block in I^n is computed, in the proposed implementation only blocks whose intensity variance is greater than a threshold T_v are computed. This approach allows to discard erroneous vectors associated with low-textured regions (regions with low intensity variance), while dramatically reducing the computational burden, as the aerial FLIR images are predominantly composed by low textured sky regions.

2.1.2. Global Motion Estimation

Global motion is estimated from $MVF^{n-1,n}$ using a restricted-affine motion model (RAM), whose parameters are estimated by means of RANSAC, a robust parameter estimation technique.

Taking into account the long distance between the camera and both targets and background in the aerial FLIR images under consideration, the projective camera model can be simplified into an orthogonal one [10]. Under this assumption, a restrictive-affine model can be used to model the ego-motion, in which only translations, rotations and zooms are considered. This motion model is defined by (2):

$$\begin{bmatrix} x^{n-1} \\ y^{n-1} \\ 1 \end{bmatrix} = \begin{bmatrix} s\cdot\cos\theta & s\cdot\sin\theta & t_x \\ -s\cdot\sin\theta & s\cdot\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x^n \\ y^n \\ 1 \end{bmatrix} \tag{2}$$

where s, θ, t_x and t_y are parameters that describe RAM, which are respectively zoom, angle of rotation, horizontal translation and vertical translation; and, x^{n-1}, y^{n-1}, x^n, y^n are the coordinates of a determined pixel in I^{n-1} and I^n respectively, which are related by RAM.

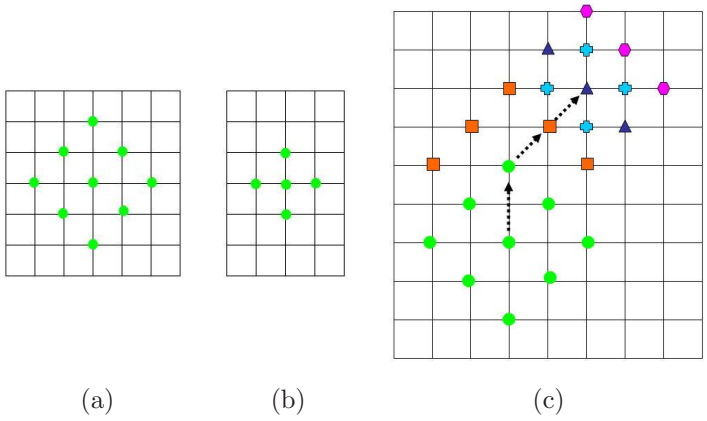


Figure 2. (a) Diamond-pattern of nine locations. (b) Diamond-pattern of five locations. (c) Diamond-pattern search method: circles corresponds to the first step, squares to the second, triangles to the third, hexagons to the fourth and crosses to the last step, where is located the minimum MAD.

RAM parameters are estimated using RANSAC, a robust iterative parameter estimation technique that tolerates up to a 50% of outliers, producing accurate estimations. As it is described in [8], RANSAC randomly selects N_{mv} motion vectors from the entire set $MVF^{n-1,n}$ to estimate RAM parameters, $\hat{P}_{RAM}$, through the Least Mean Squares algorithm (LMS). These estimated parameters, $\hat{P}_{RAM}$, are used to compute the number of inliers in $MVF^{n-1,n}$, given a tolerance threshold. If the number of inliers is greater than a threshold, T_{Nin}, the final set of RAM parameters, P_{RAM}, is computed from all inliers. Otherwise, to resolve the problem, another set of N_{mv} motion vectors is selected and the entire process is repeated up to a maximum of N_{it} times. After N_{it} iterations, if any set of RAM parameters has been computed, I^n can not be stabilized, which occurs in cases of rarely drastic displacement of the camera.

N_{mv} must be greater or equal than the dimension of the parameter space (in this case $N_{mv} \geq 2$) and to ensure that al least a set of N_{mv} motion vectors is free of outliers with a probability P_s, the number of necessary iterations is computed by (3):

$$N_{it} = \frac{\log(1 - P_s)}{\log[1 - (1 - \varepsilon^{N_{mv}})]} \tag{3}$$

where ε the maximum fraction of outliers in $MVF^{n-1,n}$.

In absence of drastic ego-motion, a set of restrictive-affine motion parameters that describes the global motion induced by the camera, P_{RAM}, is obtained for each image I^n.

2.1.3. Global Motion Compensation

This stage compensates camera motion between I^{n-1} and I^n computing the expression (2), which uses P_{RAM} for the warping. Finally, a bilinear interpolation is applied over the non-integers coordinates resulting form the warping, in order to obtained the compensated image, I_c^n.

2.2. Potential Target Detection

Potential moving target selection is accomplished in this module through a combination of segmentation techniques. First, a gradient-based segmentation, GS^n, extracts both moving and static objects of I_c^n. In parallel, a motion-based segmentation, MS^n, is carried out to obtain only some regions of moving objects between I^{n-1} and I_c^n. Then, combining GS^n and MS^n, a segmentation of moving objects is achieved, that represents a preliminary targets detection, PT^n.

2.2.1. Gradient-based Segmentation

In this stage all moving and static objects in I_c^n are segmented. First, a Canny-based edge detector is used to obtain edge image, E_c^n, from I_c^n. Then, a region extraction from edges technique is performed to obtain objects bounded by their edges, GS^n.

A Canny edge detector has been combined with an automatic gradient histogram thresholding algorithm, AGHT, [11] to obtain accurately the edges of I_c^n. AGHT determines optimum Canny hysteresis thresholds analyzing the probability distribution function of the image gradient magnitude. This follows a Rayleigh distribution assuming an additive Gaussian noise in the image, whose expression is shown in (4):

$$R(\|\nabla I_c^n\|) = \frac{\|\nabla I_c^n\|}{\sigma^2} \cdot e^{-\frac{\|\nabla I_c^n\|^2}{2\sigma^2}} \tag{4}$$

where σ is the scale parameter that is determined fitting a Rayleigh function to the image gradient histogram by means of the maximum likelihood parameter estimation technique. From this parameter, the high and low Canny hysteresis thresholds, T_H and T_L respectively, are computed as in (5):

$$\begin{aligned} T_H &= \sigma\sqrt{-2\ln\left(P_F\right)} \\ T_L &= 0.6 \cdot T_H \end{aligned} \tag{5}$$

where P_F is the acceptable proportion of false edges. As a result, E_c^n is obtained. Fig. 3 shows the Rayleigh fitting to the gradient magnitude histogram and the corresponding edge image of the FLIR image of Fig. 4(a) (the target has been bounded by a white rectangle for the sake of clarity). As it can be observed, detected edges are discontinued and do not entirely enclose objects that appear in the scene. To solve this problem, a region forming from partial edges technique [12] is used to segment the image regions that better fit to given partial edges. This technique involves the following three steps:

1. For each edge pixel, $e \in E_c^n$, to search an opposite edge pixel, $e' \in E_c^n$, which are those pixels that satisfies the condition (6) within a circular area of radius R_e:

$$\frac{\pi}{2} < |(\phi(e) - \phi(e')) \bmod (2\pi)| < \frac{3\pi}{2} \tag{6}$$

   where $\phi(e)$ is the gradient phase of the edge pixel e in the image I_c^n.

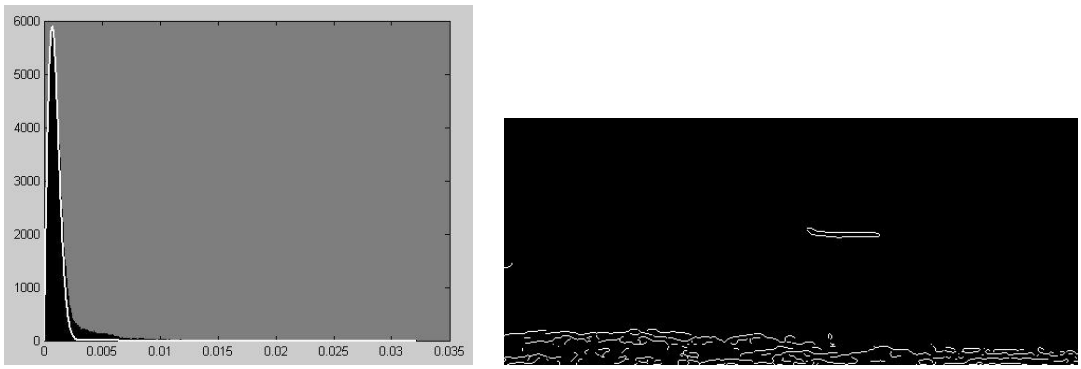
   If an opposite edge pixel is found, each pixel on the connecting straight line is marked as a potential region belonging to an object. If any opposite edge pixel is found, the next edge pixel is processed.

2. For each pixel, to compute the number of markers, $b(x)$, which represent the number of times a pixel is on the connecting line between opposite edge pixels.

3. To compute the weighted number of markers, $B(x)$, as follows (7):

$$B(x) = \begin{cases} 0.0 \text{ for } b(x) = 0 \\ 0.1 \text{ for } b(x) = 1 \\ 0.2 \text{ for } b(x) = 2 \\ 0.5 \text{ for } b(x) = 3 \\ 1.0 \text{ for } b(x) > 3 \end{cases} \tag{7}$$

   Then, a pixel x is classified as member of a region if $\sum_i B(x_i)$ in a 3×3 neighborhood of x is one or larger, otherwise it is classified as a background pixel.



(a) (b)

Figure 3. (a) Rayleigh fitting to the gradient magnitude histogram, and (b) the corresponding edge image of the FLIR image of Fig. 4(a).

Fig. 4(b) shows the output of the proposed technique applied on the edge image 3(b). Fig. 5(b) shows another example that involves very small targets, where the original FLIR image is shown in Fig. 5(a) (the targets have been bounded by a white rectangle for the sake of clarity).

Resulting of this process, GS^n is obtained, which contains all objects in I_c^n whose size is equal to or smaller than R_e.

2.2.2. Motion-based Segmentation

A motion-based segmentation is carried out through a simple automatic thresholding technique applied on the difference image computed between I^{n-1} and I_c^n.

The difference image between I^{n-1} and I_c^n is automatically thresholded analyzing its noise distribution. Additive Gaussian noise is assumed to be present in these types of FLIR images. Moreover, the difference of two images with additive Gaussian noise is another image with Gaussian noise distribution. Therefore, their noise parameters, μ_n and σ_n (mean and standard deviation respectively), can be estimated by fitting a Gaussian distribution to the difference image distribution. The threshold is automatically computed as $T_M = 3 \cdot \sigma_n$, that has associated to a false detection probability of 0.3%. After that, MS^n is obtained, a moving region segmentation which partially corresponds to moving objects in I_c^n, because of the overlapping of the areas of the own moving objects between consecutive images. Fig. 4(c) and Fig. 5(c) show the output of this stage.

2.2.3. Moving-object Detection

GS^n and MS^n segmentation are combined to only select moving objects in I_c^n. The combination is carried out by a conditional dilation [13] of MS^n using GS^n as the condition, and a 3×3 block as the structuring element. The whole set of potential moving targets are denoted as PT^n. Final results of the potential targets detection process are shown in Fig. 4(d) and Fig. 5(d).

2.3. Tracking and Target Validation

Potential targets in PT^n are tracked throughout the sequence using a template correlation technique [6] and a Kalman filter to obtain their trajectories, TJ^n, which are analyzed by means of curve fitting technique to determine whether they are congruent with an aerial target, and hence, validate them, T^n.

2.3.1. Target Tracking

Potential moving targets, PT^n, are used to carry out the tracking between consecutive images through a template correlation technique [6]. This technique creates an intensity model for each potential target in PT^{n-1} to search its best correspondence within I_c^n. This correspondence represents the temporal evolution of that potential target between consecutive frames.

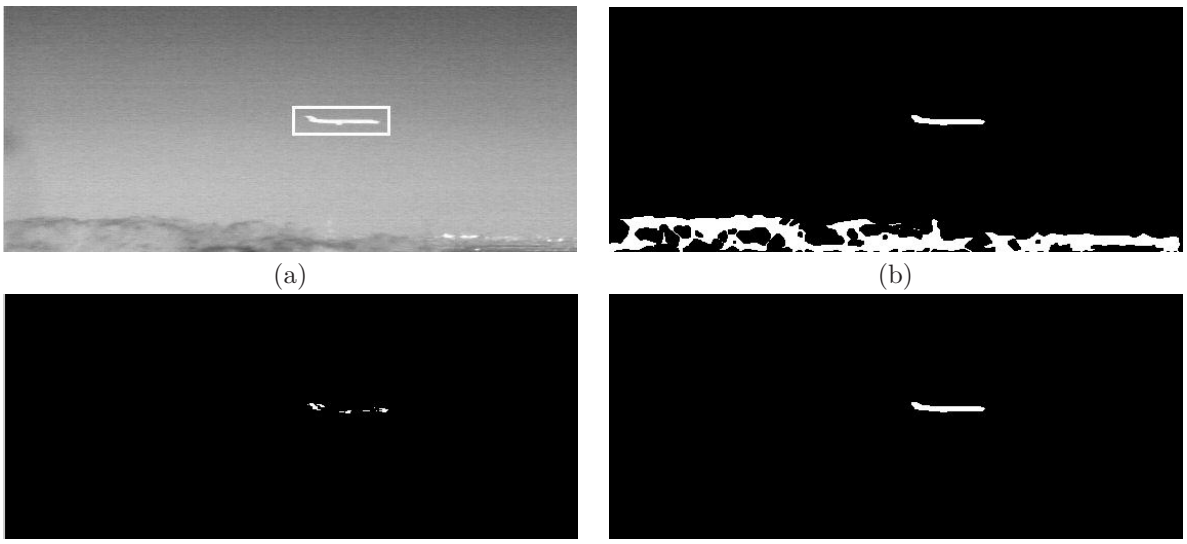


Figure 4. Stages of the *Potential Target Detection* module applied to a FLIR sequence with a medium size target: (a) FLIR image, (b) *Gradient-based Segmentation*, (c) *Motion-based Segmentation*, (d) *Moving Object Detection*.

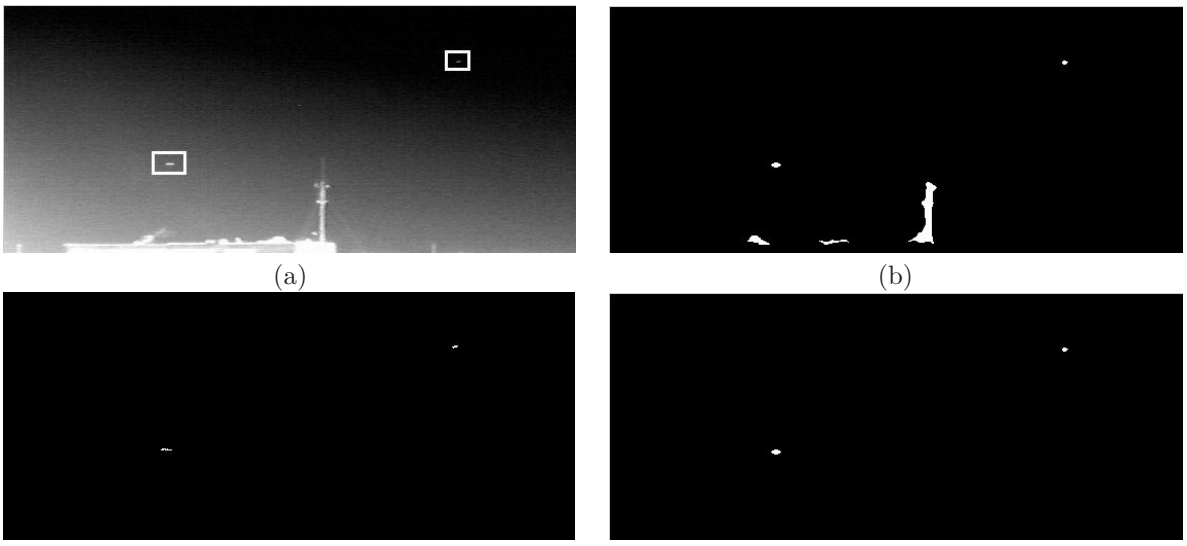


Figure 5. Stages of the *Potential Target Detection* module applied to a FLIR sequence with small size targets: (a) FLIR image, (b) *Gradient-based Segmentation*, (c) *Motion-based Segmentation*, (d) *Moving Object Detection*.

Each target model is created from a rectangular window, w^{n-1}, of dimensions $k \times l$ that circumscribes the target in I^{n-1}, and therefore, it contains the target and its surrounding background intensity information. The best correspondence with the model is calculated by minimizing the MAD between the target window in I^{n-1}, w^{n-1}, and the candidate target window in I_c^n, w_c^n, as it is shown in (8):

$$TM^n(v,z) = \frac{1}{k \cdot l} \sum_{j=1}^{l} \sum_{i=1}^{k} \left| w_c^n(i+v, j+z) - w^{n-1}(i,j) \right| \tag{8}$$

where $-l < v < l$ and $-k < z < k$.

Current and past target tracked locations are filtered by a first order Kalman filter to smooth noisy trajectories, since the expected target projected trajectories can be approximated by linear steps. As a result, a set of potential target filtered trajectories are obtained, $TJ^n = \{TJ_1^n, ..., TJ_i^n, ..., TJ_N^n\}$, where N is the number of potential tracked targets and each potential target trajectory can be expressed as $TJ_i^n = \{r_i^m, ..., r_i^n\}$, where r_i^m is the i target location in the temporal instant m, in which the target appeared for the first time.

2.3.2. Target Validation

Each potential target trajectory, TJ_i^n, is analyzed to test if it is consistent with a real target trajectory, in order to validate it, discarding those potential targets associated to erratic or non linear-wise trajectories produced by inherent drawbacks of FLIR images and birds.

The performed analysis consists in evaluating the number of outlier trajectory locations, L_{out}, which are computed through Median Absolute Deviation technique (MedAD). If L_{out} is larger than a predefined threshold Th_{out} the potential target is discarded, otherwise the potential target is validated.

MedAD [14] is a scale parameter estimator that automatically computes the number of outliers in TJ_i^n as follows (9):

$$L_{out} = \left\{ r_i^j \in TJ_i^n | \left(R_i^j\right)^2 > \left(2.5 \cdot \hat{\beta}\right)^2 \right\} \tag{9}$$

where $\left(R_i^j\right)^2$ is the square residual distance between a trajectory location and its corresponding fitted trajectory location. This has been computed using a line fitting model and Least Median Squares algorithm [12], that robustly estimates the parameters of the line that better fits to the trajectory. And $\hat{\beta}$ is the inlier scale estimator given by (10):

$$\hat{\beta} = 1.4826 \cdot \left(1 + \frac{5}{(n-m+1)-2}\right) \cdot \sqrt{\text{median}\left\{\left(\mathrm{R}_i^j\right)^2\right\}} \tag{10}$$

where $(n-m+1)$ is the number of trajectory locations in TJ_i^n.

As the final result from this trajectory filtering, only those targets which follow consistent trajectories are validated, T^n.

3. RESULTS

The system has been tested with real FLIR sequences captured by an interlaced gray-level infrared camera in 8-12 μm range with a resolution of 512×512 pixels. The camera was mounted on a moving platform that produced a global motion in the sequences. These sequences are composed by large non-textured sky regions combined with cloud and earth image areas that represent the sources of clutter. In addition, sequences are affected by varying illumination conditions.

Fig. 6 shows a real FLIR image in (a) and two different motion vector field estimations in (b) and (c). The motion vector field in (b) has been computed for all the blocks in the image, which produces a huge amount of

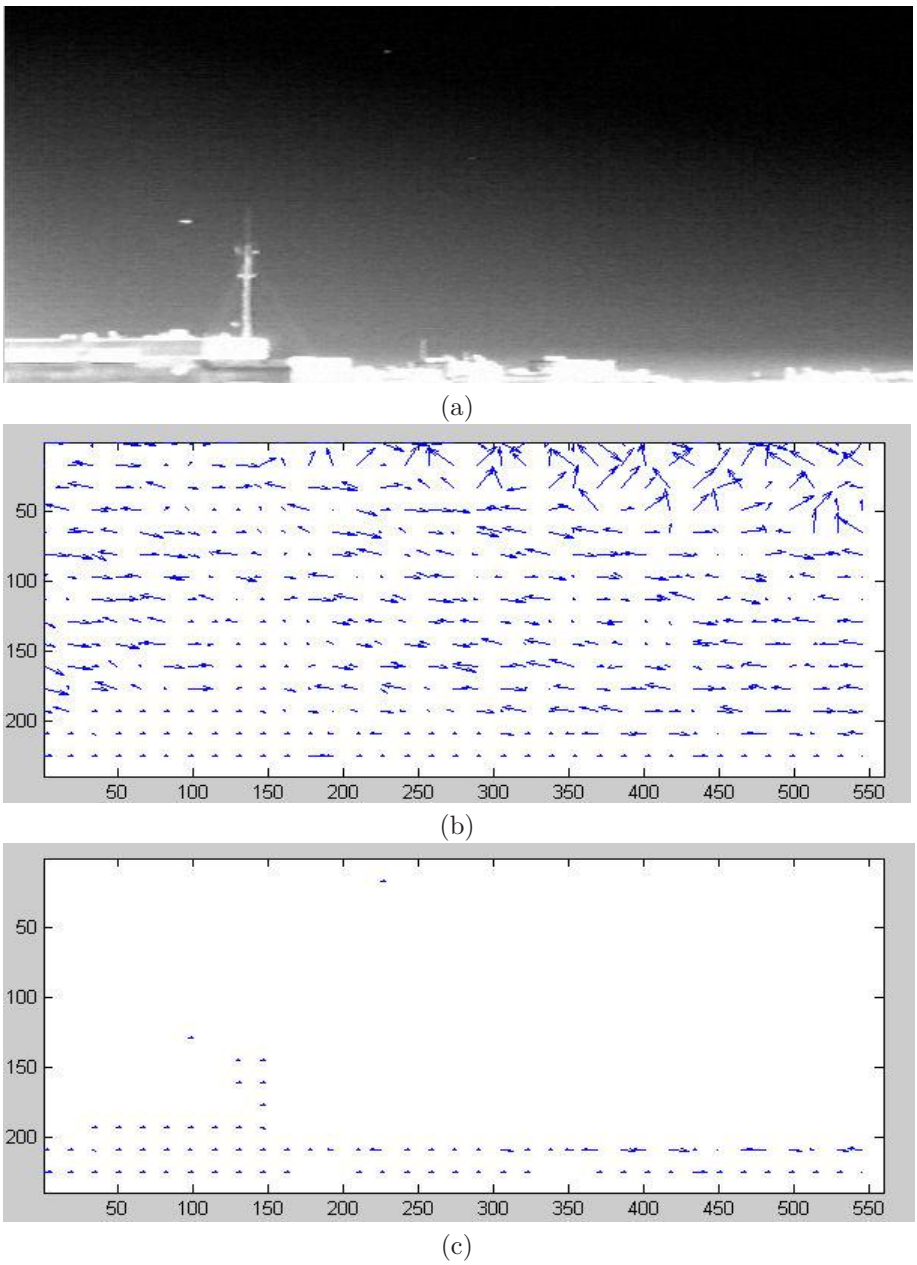


Figure 6. (a) Real FLIR image, (b) motion vector field for all blocks in the image, and (c) motion vector field for blocks with large intensity variance.

outlier motion vectors due to the aperture problem present those blocks that lie on non-textured regions. On the contrary, the motion vector field in (c) has been only computed for those blocks holding textured details (blocks whose variance is larger than a predefined threshold as presented in 2.1.1). The result is that only a few outliers are present in the $MVF^{n-1,n}$, allowing to obtain accurate global motion estimations.

The performance of the global motion compensation is presented in Fig. 7. It shows the global motion compensation error measure computed as the Peak Signal to Noise Ratio (PSNR) between I^{n-1} and I_c^n for two different situations: global motion compensation using RANSAC, (represented in Fig. 7 by a solid line); and that obtained applying only Least Mean Squares (LMS) (represented in Fig. 7 by a dashed line). As it can be

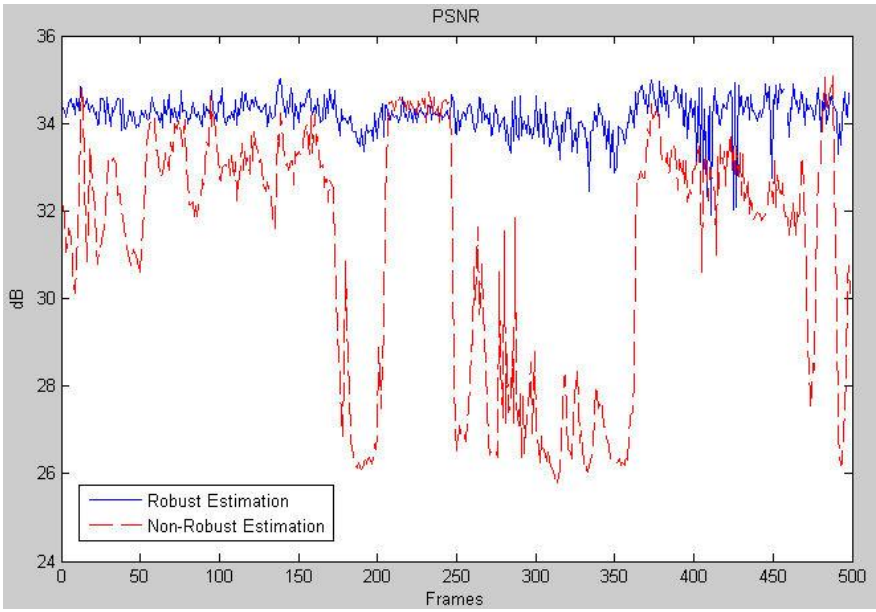


Figure 7. Graphic about PSNR measures: the solid line corresponds to robust parameter estimation and the dashed line to non-robust parameter estimation.

observed, RANSAC offers better results since it is able to detect outliers and discard them, while LMS is very sensitive to outliers. Moreover, RANSAC obtains results much more stable throughout the entire sequence than those obtained by LMS.

The *Tracking and Target Validation* module has been tested in FLIR sequences containing real and false targets. Fig. 8 shows a graphic that contains trajectory locations of a real target (marked with crosses), Kalman-filtered trajectory locations (marked with diamonds joint with a dashed line), the best fitting line (marked with a solid line) and the boundaries that separate inliers from outliers (marked with dashed lines). Notice that all filtered trajectory locations lies into the boundaries, i.e. the number of outliers is zero, and therefore, the potential target is validated as a real target.

Fig. 9 shows a similar graphic but associated to a false target. In this case, the number of outliers is larger than zero, and hence, the potential target is discarded.

The performance of the whole system has been measured by means of the detection and false alarm rates over a set of real FLIR sequences, in which always exist reduced regions of clutter (earth o clouds), necessary for camera ego-motion compensation. An excellent Detection Rate of 99.2% has been achieved. False alarm rate has been acquired before and after *Tracking and Target Validation* module to observe its improvement: false alarm rate goes from 13.7% down to 0.4% when using the proposed trajectory filtering strategy.

4. CONCLUSIONS

A novel strategy for improving the false alarm rate in an ATDT system for aerial targets in airborne FLIR imagery has been presented in this paper. This strategy consists in an efficient target trajectory filtering that refines the detection process. Furthermore, the computational complexity of the system, especially that associated to the ego-motion compensation task, has been carefully reduced in order to reach a real-time operation. The system estimates local motion through a very fast selective-search block-matching algorithm. Global motion is robustly inferred from local motion using a restricted-affine motion model, and a combination of gradient and motion segmentation techniques are used to detect potential aerial targets from each compensated image. Potential aerial targets are tracked by means of a target template-based correlation technique, followed by a Kalman filter smoothing process. The smoothed trajectories are analyzed through a robust curve fitting technique to filter

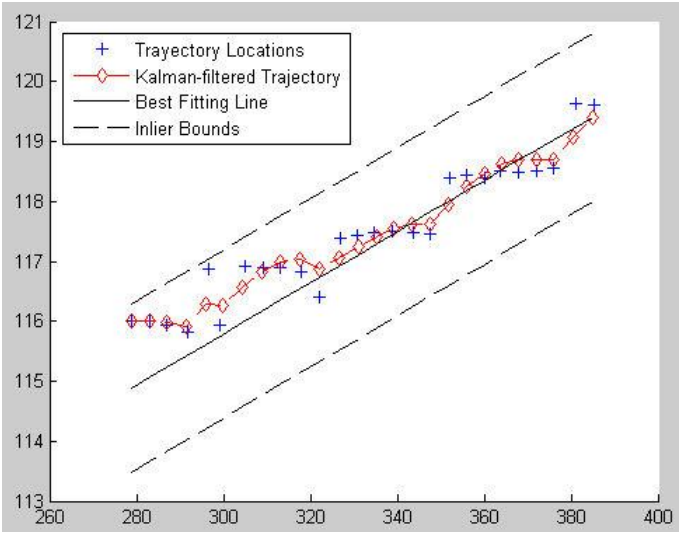


Figure 8. Graphic with trajectory locations of a real target (crosses), Kalman-filtered trajectory locations (diamonds joined with a dashed line), the best fitting line (solid line) and the boundaries that separate inliers from outliers (dashed lines).

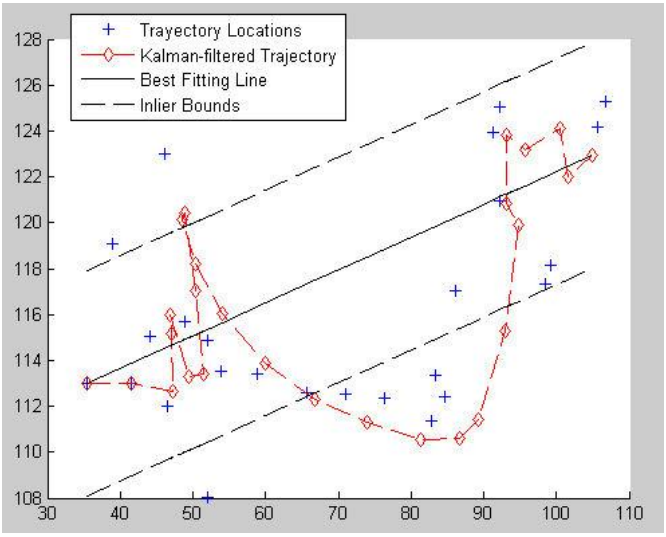


Figure 9. Graphic with trajectory locations of a false target (crosses), Kalman-filtered trajectory locations (diamonds joined with a dashed line), the best fitting line (solid line) and the boundaries that separate inliers from outliers (dashed lines).

those inconsistent with the expected trajectory for a real target. This approach allows to reduce significantly the false alarm rate caused by inherit drawbacks of FLIR imagery such as artifacts, changes in illumination and variations in thermal signature, as well as, moving clutter due to birds. Presented results show the good performance of this strategy.

ACKNOWLEDGMENTS

This work has been partially supported by the Ministerio de Ciencia y Tecnología of the Spanish Government under project TIN2004-07860 (Medusa) and by the Comunidad de Madrid under project P-TIC-0223-0505 (Pro-Multidis).

REFERENCES

1. A. Strehl and J. K. Aggarwal, "Detecting moving objects in airborne forward looking infra-red sequences," in *Workshop on Computer Vision Beyond Visible Spectrum*, *Proc. IEEE*, pp. 3–12, 1999.
2. A. Strehl and J. Aggarwal, "Modeep: a motion-based object detection and pose estimation method for airborne flir sequences," *Machine Vision and Applications* **11(6)**, pp. 267–276, 2000.
3. E. Estalayo, L. Salgado, F. Jaureguizar, and N. García, "Efficient image stabilization and automatic target detection in aerial flir sequences," in *Automatic Target Recognition XVI*, *Proc. SPIE* **6234**, p. 62340N, 2006.
4. A. Yilmaz, K. Shafique, N. Lobo, X. Li, T. Olson, and M. A. Shah, "Target-tracking in flir imagery using mean-shift and global motion compensation," in *Workshop on Computer Vision Beyond Visible Spectrum*, *Proc. IEEE*, 2001.
5. A. Yilmaz, K. Shafique, and M. Shah, "Target tracking in airborne forward looking infrared imagery," *Image and Vision Computing Journal* **21(7)**, pp. 623–635, 2003.
6. A. Bal and M. Alam, "Automatic target tracking in flir image sequences using intensity variation function and template modeling," *IEEE Transactions on Instrumentation and Measurement* **54(5)**, pp. 1846–1852, 2005.
7. G. Wolberg, *Digital Image Warping*, IEEE Computer Society Press, 1994.
8. C. Stewart, "Robust parameter estimation in computer vision," *SIAM Reviews* **41(3)**, pp. 513–537, 1999.
9. E. Chan and S. Panchanathan, "Review of block matching based motion estimation algorithms for video compression," *Canadian Conference on Electrical and Computer Engineering* **1**, pp. 151–154, 1993.
10. W. Meier and H. Stein, "Estimation of object and sensor motion in infrared image sequences," in *Int. Conf. on Image Processing*, *Proc. IEEE* **1**, pp. 568–572, 1994.
11. P. Rosin, "Edges: Saliency measures and automatic thresholding," *Machine Vision and Applications* **9(4)**, pp. 139–159, 1999.
12. M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, Thomson Learning, 1998.
13. E. R. Dougherty, *An introduction to morphological image processing*, SPIE Optical Engineering Press, 1992.
14. P. Meer, C. Stewart, and D. Tyler, "Robust computer vision: an interdisciplinary challenge," *Computer Vision and Image Understanding* **78(1)**, pp. 1–7, 2000.
15. S. Young, H. Kwon, S. Z. Der, and N. M. Nasrabadi, "Adaptive target detection in forward-looking infrared imagery using the eigenspace separation transform and principal component analysis," *Proc. Optical Engineering* **43**, pp. 1767–1776, 2004.
16. H. Seok and J. Lyou, "Digital image stabilization using simple estimation of the rotational and translational motion," in *Acquisition, Tracking and Pointing XIX*, *Proc. SPIE* **5810**, pp. 170–181, 2005.
17. M. Irani, B. Rousso, and S. Peleg, "Recovery of ego-motion using region alignment," *Trans. on Pattern Analysis and Machine Intelligence* **19(3)**, pp. 268–272, 1997.
18. J. Chang, W. Hu, M. Chang, and B. Chang, "Digital image translational and rotational motion stabilization using optical flow technique," *Trans. on Consumer Electronics* **48(1)**, pp. 108–115, 2002.
19. C. Marimato and R. Chellappa, "Evaluation of image stabilization algorithms," in *Int. Conf. on Accoustics, Speech, and Signal Processing*, *Proc. IEEE* **5**, pp. 2789–2792, 1998.